

## SERVER\_SOURCE

```

readme
1      Connectivity between the various Perl files

2      (1) ci.pl
3          project.pl
4          host.pl
5          getsessionid.pl
6          project.pl
7          webdate.pl
8      -> acceptdecline.pl (by way of form submission)

9      acceptdecline.pl
10     cgi.pl
11     getsessionid.pl
12     project.pl ...
13     -> (2a) declined.html OR
14     -> (2b) ci.pl

15     (3) save.pl
16     getsessionid.pl

17     (4) calculate.pl
18     project.pl ...
19     getsessionid.pl ...
20     -> (4a) overuse.html OR
21     -> (4b) getresults.pl (by way of META refresh)

22     (5) getresults.pl
23     getsessionid.pl ...
24     calculate2.pl
25     project.pl ...
26     getsessionid.pl ...
27     readstuff.pl
28     project.pl (!)
29     getsessionid.pl (!)
30     ansoft_files.pl
31     remoteTLINE.pl
32     getheadpost.pl

```

## all\_md5

```

1      MD5 checksums
2      c4616dce3206bfba1e1d26e910fcda19  readme
3      f627deb66ff95612e34bbc2840b44069  all_md5
4      2f73d693d13e2de292c2a9a17b09bb40  acceptdecline.pl
5      d4d66bd78c7e3d244a844dc8c6568d83  calculate.pl
6      4f0ef466c56dbf48f431045b3c6b65af  calculate2.pl
7      a23e93229c5cd96f44ed6bfa8cb43923  cgi.pl
8      3dd2bd09c6af821ac9489471e0140b51  ci.pl
9      5f6136da8fbc267052ed9a6elfcd86c4  getresults.pl
10     7889f9cab3787db2b0a4231ec128a331  getsessionid.pl
11     a926a04944ceeeecaf8c5ba09675aa46  host.pl
12     c04024bba78bf2e5da1fc944f3c083e2  makeauth.pl
13     ccf4d0e44f6b7b5fa920e23ae022d46b  project.pl
14     f62c02714876d4686de1139739abccdf  save.pl
15     d30a1ca9731e7feccf5922c1681192dc  webdate.pl
16     c2898185d40e61d3e8f98add70d26f4a  native_CrossSection.html
17     2daeb6d25d7d4ffe52ed835721512f56  nav3_CrossSection.html
18     cf20dc1f37dc1c1012fdf7fccb8117d1  ../../html/Stackup/declined.html

```

```

19      90d33ac8912a6b25c3332b34dec14e71  ../../html/Stackup/error.html
20      cd1410f9391b66a64abe3f0368f72e66  ../../html/Stackup/overuse.html

```

```
acceptdecline.pl
```

```

1      #!/usr/bin/perl
2
3      use strict;
4
5      my $debug;
6      # $debug = 1;
7      if ($debug) { print "Content-Type: text/html\n\n<HTML><BODY><PRE>"; }
8
9      require("cgi.pl");
10
11     # GLOBALS
12     use vars qw( $debug );
13
14     require("getsessionid.pl");
15     use vars qw( $suid $user_id $usage $path_info $save_copy $cookie);
16
17     my %values = &process_form();
18     my $time = hex( &obscure( substr($values{"token"},0,8),
19                               substr($values{"token"},8,8) ) );
20     my $delay = $time+15-time();
21     if ($debug) { print "time: $time delay: $delay\n"; }
22
23     require("project.pl");
24
25     use vars qw($HOST_IP);
26
27     use vars qw($PROJECT $USAGE_MAX $USAGE_INTERVAL $RECORD_LENGTH
28 $NUM_RECORDS);
29
30     if ($values{"choice"} =~ /ACCEPT/) {
31         # if ($delay > 0) { sleep($delay); }
32
33         print "Location: http://$HOST_IP/cgi-bin/$PROJECT/ci.pl\n";
34         print "Pragma: no-cache\nExpires: Mon, 01 Jul 1996 00:00:00 GMT\n";
35         if ($delay > -300) { print $cookie; }
36         print "\n";
37     }
38     else {
39         print "Location: http://$HOST_IP/$PROJECT/declined.html\n";
40         print "Pragma: no-cache\nExpires: Mon, 01 Jul 1996 00:00:00 GMT\n";
41         print "\n";
42     }
43 }

```

```
calculate.pl
```

```

1      #!/usr/bin/perl
2      # calculate.pl
3      # Copyright (C) 1999-2000 Rode Consulting, Inc.
4      # All rights reserved.
5
6      use strict;
7
8      # GLOBALS
9      use vars qw( $debug );
10     # $debug = 1;

```

```

9      require("project.pl");

10     use vars qw($HOST_IP);
11     use vars qw($PROJECT $USAGE_MAX $USAGE_INTERVAL $RECORD_LENGTH
$NUM_RECORDS);

12     require("getsessionid.pl");
13     use vars qw ( $suid $new_suid $user_id $timestamp $usage $valid
$spath_info $cookie);

14     $USAGE_MAX=10;
15     if ($usage >= $USAGE_MAX) {
16         print "Location: http://$HOST_IP/$PROJECT/overuse.html\n";
17         print "\n";
18         exit(0);
19     }

20     #print "Location: http://", $HOST_IP, "/cgi-bin/$PROJECT/calculate2.pl\n";
21     #print "\n";
22     #
23     # Rather than directly forward to the script that does the real
calculation
24     # We instead return a small HTML page that displays a message, before
25     # forwarding to the final location.

26     my $hash = &hash($user_id);

27     # Write new cookie to cookie cache
28     sysopen(AUDIT_DB, "auth.dat", 2) or die("Can't open authorization
database!");
29     sysseek( AUDIT_DB, $hash*$RECORD_LENGTH, 0 );
30     my $written = syswrite( AUDIT_DB, ($new_suid."\n"),
31                             $RECORD_LENGTH );

32     die "System write error: $!\n" unless defined $written;
33     close(AUDIT_DB);

34     print "Pragma: no-cache\nExpires: Mon, 01-Jul-1996 00:00:00 GMT\n";
35     print $cookie;

36     print "Content-Type: text/html\n\n";
37     print <<"EOF"
38     <HTML><HEAD><TITLE>Extraction in progress...</TITLE>
39     <META HTTP-EQUIV="refresh" CONTENT="0;URL=http://$HOST_IP/cgi-
bin/$PROJECT/getresults.pl">

40     <SCRIPT LANGUAGE="JavaScript"><!--
41         badbrowser = ((navigator.appName == "Microsoft Internet Explorer")
&& (parseInt(navigator.appVersion) == 4) && (navigator.appVersion.indexOf("Mac") >
0));
42         if (badbrowser) this.focus();
43         if (!badbrowser && window.focus) this.focus();
44         //--></SCRIPT>

45     </HEAD>
46     <BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF" VLINK="#AA00AA"
ALINK="#FF0000">

```

```

47      <TABLE VALIGN=CENTER WIDTH="100%" HEIGHT="100%"><TR><TD VALIGN=CENTER
ALIGN=CENTER>
48      <CENTER><H2>Please wait...<BR>Extraction in progress</H2></CENTER>
49      </TD></TR></TABLE>
50      </BODY>
51      </HTML>

52      EOF
53      ;
54      1;

calculate2.pl
1      #!/usr/bin/perl
2      # calculate2.pl
3      # Copyright (C) 1998-2000 Rode Consulting, Inc. (RCI)
4      # All rights reserved.

5      use strict;

6      # GLOBALS
7      use vars qw( $debug );
8      #$debug = 1;

9      if ($debug) {
10         print "Content-Type: text/html\n";
11         print "Pragma: no-cache\nExpires: Mon, 01 Jul 1996 00:00:00 GMT\n";
12         print "\n";
13
14         print "<BODY><PRE>\n";
15     }

16     require("project.pl");

17     use vars qw($HOST_IP);
18     #use vars qw($DOMAIN_NAME $HOST_NAME $HOST_IP $HOST_PREFIX
$HOST_CGI_PREFIX);
19     #use vars qw($HOST_PREFIX_JAVA $HOST_CGI_PREFIX_JAVA);
20     #use vars qw($BODY $NOCACHE_HEADER $GS $JP2GIF $ALCHEMY);
21     #use vars qw(%HTML_MACROS);

22     use vars qw($PROJECT $USAGE_MAX $USAGE_INTERVAL);

23     require("getsessionid.pl");
24     use vars qw( $suid $user_id $timestamp $usage $valid $path_info
$cookie);

25     use vars qw( @items $user_id $root $dont_do_it $results);
26     use vars qw( $units_name $units $scale );

27     my $LOCK_SH=1; my $LOCK_EX=2; my $LOCK_NB=4; my $LOCK_UN=8;

28     if (!$dont_do_it && ($usage > $USAGE_MAX)) {
29         print "Location: http://$HOST_IP/$PROJECT/overuse.html\n";
30         print "\n";
31         exit(0);
32     }

33     if (!$dont_do_it && ((time()-$timestamp) > 15)) {
34         print "Location: http://$HOST_IP/$PROJECT/error.html\n";

```

```

35     print "\n";
36     exit(0);
37 }

38 # Retrieve the binary data from a file
39 if ( $dont_do_it ||
40     ($valid>0 && -e "tmp/stuff".$user_id.sprintf("%04d",$usage)) ) {

41     # require("readstuff.pl");
42     # $root = "x.pjt";
43     # require("field_extractor_files.pl");

44     # Early return for use with ez2dzip.pl
45     # if ($dont_do_it) { return 1; }

46     # open(LOCKFILE,">>tmp/lockfile");
47     # flock(LOCKFILE, $LOCK_EX);
48     # seek(LOCKFILE, 0, 2);
49     # require("remoteTLINE.pl");
50     # flock(LOCKFILE, $LOCK_UN);
51     # close(LOCKFILE);

52     open(SAVE,">tmp/results".$user_id.sprintf("%04d",$usage));
53     print SAVE "DUMMY";
54     close SAVE;
55 }
56 else {
57     print "Location: http://$HOST_IP/$PROJECT/error.html\n\n";
58     exit 0;
59 }
60 1;

cgi.pl
1  #!/usr/local/bin/perl
2  #Copyright (C) 1996-8 Rode Consulting, Inc. All rights reserved.

3  #package myCGI;

4  use strict;

5  # GLOBALS:
6  # $cgi_query: The query string (source depends on
REQUEST_METHOD)
7  # (logged by other scripts)
8  # TEMP %form_values is global for compatibility

9  # Special (evil?) mode flags:
10 # $debug: Global cross-module debug flag
11 # $_cgi_lowercase: Form value names are all lowercased

12 use vars qw($debug $_cgi_lowercase $cgi_query %form_values);

13 if ($debug) { print "cgi.pl called\n"; }

14 # process_form()
15 sub process_form {

16     my ($valid,@pairs) = ("",(""));

```

```

17         $| = 1;
18     if ($debug) {        print "Method: ", $ENV{'REQUEST_METHOD'}, "<BR>\n";
19     }
20     $valid = "";
21     if ($ENV{'REQUEST_METHOD'} eq 'GET')
22     {
23         $cgi_query = $ENV{'QUERY_STRING'};
24         $valid = ($ENV{'QUERY_STRING'}) ? "true" : "";
25     }
26     elsif ($ENV{'REQUEST_METHOD'} eq 'POST')
27     {
28         $valid = ($ENV{'CONTENT_LENGTH'} > 0) ? "true" : "";
29         if ($valid) { read(STDIN, $cgi_query, $ENV{'CONTENT_LENGTH'}); }
30         $cgi_query =~ s/\n/\&/g;
31     }
32     elsif ($debug) { print "Unrecognized method:
33     $ENV{'REQUEST_METHOD'}\n"; }
34
35     if ($debug) {        print "CGI arguments: ", $cgi_query, "<BR>\n";
36     }
37
38     if ($valid)
39     {
40         @pairs = split(/&/, $cgi_query);
41         foreach my $pair (@pairs)
42         {
43             $pair =~ s/\+/\+/g;
44             $pair =~ s/\+/\ /g;
45             my ($name, $value) = split(/=/, $pair);
46             $name =~ s/%([a-fA-F0-9]{2})/pack('c',hex($1))/eg;
47             $value =~ s/%([a-fA-F0-9]{2})/pack('c',hex($1))/eg;
48             $value =~ s/%2[Ff]/\//g;
49             if ($_cgi_lowercase) { $name =~ tr/A-Z/a-z/; }
50             if ($form_values{$name})
51             { $form_values{$name} .= "&" . $value; }
52             else
53             { $form_values{$name} = $value; }
54
55             if ($debug) {        print $name, " = '", $form_values{$name},
56             "'\n"; }
57         }
58     }
59     %form_values;
60 }
61
62 # get_cookie( -cookiename- )
63 # Returns the first cookie by the given name
64
65 sub get_cookie {
66     my ($cookie) = @_ ;
67     my (@cookies, $cookieName, $cookieVal) = ("","","");
68
69     if ($debug) { print "HTTP_COOKIE: ", $ENV{"HTTP_COOKIE"}, "\n"; }
70     if ($ENV{"HTTP_COOKIE"} =~ /$cookie/)
71     {
72         @cookies = split(/;\s*/, $ENV{"HTTP_COOKIE"});

```

```

63         if ($debug) { print '@cookies: ',@cookies,"\n"; }
64
65         foreach (@cookies)
66         {
67             if ($_ =~ /$cookie/)
68             {
69                 ($cookieName, $cookieVal) = split (/=/,$_);
70                 return($cookieVal);
71             }
72         }
73     }
74     else { return(''); }
75 }

76 ##@LOOKUP = split(/Name:/,'nslookup $ENV{"REMOTE_ADDR"}');
77 ##@LOOKUP = split(/Address:/, $LOOKUP[1]);
78 ##$client_name = $LOOKUP[0];
79 #$client_name = $ENV{"REMOTE_ADDR"};
80 #$client_name =~ s/ //g;

81 #package form;
82 1;

ci.pl
1    #!/usr/bin/perl
2
3    use strict;
4
5    # GLOBALS
6    use vars qw( $debug );
7
8    my $debug;
9    #$debug = 1;
10   if ($debug) { print "Content-Type: text/html\n\n<HTML><BODY><PRE>"; }
11
12   require("project.pl");
13   use vars qw($DOMAIN_NAME $HOST_NAME $HOST_IP $HOST_PREFIX
14   $HOST_CGI_PREFIX);
15
16   use vars qw($PROJECT $USAGE_MAX $USAGE_INTERVAL $RECORD_LENGTH
17   $NUM_RECORDS);
18
19   require("getsessionid.pl");
20   use vars qw( $suid $user_id $usage $path_info $save_copy $cookie);
21
22   if ($debug) {
23       print $suid," ",$user_id," ",$usage," ",$path_info," ",$cookie;
24   }
25
26   print "Content-Type: text/html\n";
27   print "Pragma: no-cache\nExpires: Mon, 01 Jul 1996 00:00:00 GMT\n";
28   print "\n";
29
30   if (length($suid) > 0) {
31       if ( !($ENV{"HTTP_USER_AGENT"} =~ /3.0/) ) {
32           open(HTML, "native_CrossSection.html");
33       }
34       else {
35           open(HTML, "nav3_CrossSection.html");
36       }
37   }

```

```

24         read(HTML, my $html, 1000000);
25         close(HTML);

26         $html =~ s/<HOST_IP>/$HOST_IP/g;

27         print $html;
28     }
29     else { # New cookie

30         open(DISCLAIMER, "../../html/netsim/termsfuse.html");
31         read(DISCLAIMER, my $disclaimer, 1000000);
32         close(DISCLAIMER);

33         my $time = sprintf("%08x",time());
34         my $mask = substr(&sumcheck($time),0,8);
35         my $token = &obscure($time,$mask).$mask;
36         $disclaimer =~ s/<!-- marker1 -->/ \ (Please read carefully then
click <B>ACCEPT<\> or <B>DECLINE<\> at the bottom of this page\)/;

37         my $replacement = <<"EOF"

38         <CENTER>
39         <APPLET CODE="AcceptDecline.class" CODEBASE="http://$HOST_IP/$PROJECT/"
WIDTH=250 HEIGHT=40>
40         <PARAM NAME=token VALUE="$token">
41         </APPLET>
42         </CENTER>
43         EOF
44         ;
45         $disclaimer =~ s/<!-- marker2 -->/ $replacement/;

46         print $disclaimer;
47     }

48     exit(0);

getresults.pl
1     #!/usr/bin/perl

2     use strict;

3     require("getsessionid.pl");
4     use vars qw( $debug );
5     use vars qw ( $suid $new_suid $user_id $usage $path_info $save_copy
$cookie);

6     my $results_filename = "tmp/results".$user_id.sprintf("%04d",$usage);

7     if (! (-e $results_filename) ) {
8         open(SAVE,">$results_filename");
9         close(SAVE);
10        require "calculate2.pl";
11    }

12    print "Content-Type: text/html\n";
13    print "Pragma: no-cache\nExpires: Mon, 01 Jul 1996 00:00:00 GMT\n";
14    print "\n";
15    print "<HTML><HEAD>\n";
16    print "<TITLE>Field Extraction Results</TITLE>\n";

```



```

17     print "<SCRIPT LANGUAGE=\"JavaScript\"><!--\n";
18     print "        badbrowser = ((navigator.appName == \"Microsoft Internet
Explorer\") && (parseInt(navigator.appVersion) == 4) &&
(navigator.appVersion.indexOf(\"Mac\") > 0));\n";
19     print "        if (badbrowser) this.focus();\n";
20     print "        if (!badbrowser && window.focus) this.focus();\n";
21     print "//--></SCRIPT>\n";
22     print "<STYLE TYPE=\"text/css\">\nBODY    { font-size:10pt }\n    H1
    { font-size:24pt }\n    H2    { font-size:20pt }\n    H3    { font-size:16pt
}\n    TABLE { font-size:10pt }\n    TD    { font-size:10pt }\n</STYLE>\n";
23     print '</HEAD>';
24     print '<BODY BGCOLOR=\"#FFFFFF\" TEXT=\"#000000\" LINK=\"#0000FF\"
VLINK=\"#AA00AA\" ALINK=\"#FF0000\">','\n";

25     my $results;
26     open(STUFF,$results_filename);
27     read(STUFF,$results,1000000);
28     close(STUFF);

29     if ($results =~ /ERROR/ || length($results) == 0) {
30         print "<CENTER>\n";
31         print "<TABLE><TR><TD WIDTH=\"80%\">\n";
32         print "<BR><H3>We're sorry, but an error occurred attempting to
extract this geometry.<BR> Please check for overlapping elements and try
again.</H3>\n";
33         print "</TD></TR></TABLE>\n";
34         print "</CENTER>\n";
35         print "</BODY></HTML>\n";
36         exit 1;
37     }
38     else {

39         # POST PROCESS RESULTS INTO HTML (THIS IS A PLACEHOLDER)
40         print "<CENTER>\n";
41         print "<H3>Complete</H3>";
42         print "</CENTER>\n";
43         print "</BODY></HTML>\n";
44         exit 0;
45     }

46     sub pretty() {
47         my $precision = pop(@_);
48         my $number = pop(@_);
49         my $index = index $number,"e",0;
50         if ($index < 0) { $index = index $number,"E",0; }
51         if ($index>0 && $index > $precision) {
52             return substr($number,0,$precision).substr($number,$index);
53         }
54         else { if ($index>0) { return $number; }
55               else { return substr($number,0,$precision); }
56         }
57     }

getsessionid.pl
1     #!/usr/bin/perl
2     #Copyright (C) 1997-2000 Rode Consulting, Inc.
3     #All rights reserved.

4     use strict;

```

```

5      use MD5;

6      require("project.pl");

7      use vars qw($DOMAIN_NAME $HOST_NAME $HOST_IP $HOST_PREFIX
$HOST_CGI_PREFIX);
8      use vars qw($HOST_PREFIX_JAVA $HOST_CGI_PREFIX_JAVA);
9      use vars qw($BODY $NOCACHE_HEADER $GS $JP2GIF $ALCHEMY);
10     use vars qw(%HTML_MACROS);

11     use vars qw($PROJECT $USAGE_MAX $USAGE_INTERVAL $NUM_RECORDS
$RECORD_LENGTH);

12     require("webdate.pl");

13     # GLOBALS
14     use vars qw( $debug );
15     use vars qw( $suid $new_suid $user_id $timestamp $usagestamp $usage
$valid $path_info $cookie);

16     my $COOKIE_LENGTH = 35;

17     #$debug = 1;
18     if ($debug) {
19         print "Content-type: text/html\n\n<HTML><BODY><PRE>\n";
20         print "path_info: $ENV{'PATH_INFO'}\ncookie: $ENV{'HTTP_COOKIE'}\n";
21     }

22     # Any old random number will do (32 bytes)...
23     my $local_cryptkey =
"baceeed5e3ff05b81b3688c1e10b914bd2a1518edb6c9090358eb21cce6da82";

24     sub obscure() {
25         my $mask = pop @_;
26         my $target = pop @_;

27         my $obscured = "";
28         for (my $i=0; $i<length($target); $i++) {
29             $obscured .= sprintf( "%1x", hex(substr($target,$i,1)) ^
hex(substr($mask,$i,1)) ^ hex(substr($local_cryptkey,$i,1)) );
30         }

31         return $obscured;
32     }

33     sub sumcheck() {
34         my $source = pop @_;

35         # my $sum = `echo $source | md5sum 2>/dev/null`;
36         # $sum =~ s/\s*-\s*//g;
37         my $md5 = new MD5;
38         $md5->add($source."\n");
39         my $sum = unpack( "H32", $md5->digest() );

40         return $sum;
41     }

42     sub encode_suid() {
43         my $usage = pop @_;

```

```

44         my $usagestamp = pop @_ ;
45         my $timestamp = pop @_ ;
46         my $user_id = pop @_ ;

47         #open(LOGFILE,">>logfile");
48         #print LOGFILE "E $user_id $timestamp $usagestamp $usage\n";
49         #close(LOGFILE);

50         my $combined =
$user_id.sprintf("%08x%08x%04d",$timestamp,$usagestamp,$usage);
51         my $sum = &sumcheck($combined);

52         $sum =~ s/^0*/g;
53         $sum = substr($sum,$sum,0,$COOKIE_LENGTH);

54         my $obscured = &obscure($combined,$sum);
55         return $obscured.$sum;
56     }

57     sub decode_suid() {
58         my $encoded = pop @_ ;

59         my $unobscured =
&obscure(substr($encoded,0,$COOKIE_LENGTH),substr($encoded,$COOKIE_LENGTH,$COOKIE_LE
NGTH));

60         my $user_id = substr($unobscured,0,15);
61         my $timestamp = hex substr($unobscured,15,8);
62         my $usagestamp = hex substr($unobscured,23,8);
63         my $usage = hex substr($unobscured,31,4);

64         #open(LOGFILE,">>logfile");
65         #print LOGFILE "D $user_id $timestamp $usagestamp $usage\n";
66         #close(LOGFILE);

67         my $valid = 0;
68         if ( $encoded eq &encode_suid($user_id, $timestamp, $usagestamp,
$usage) ) { $valid = 1; }

69         return ($user_id, $timestamp, $usagestamp, $usage, $valid);
70     }

71     # GET USER_ID, USAGE AND PATH_INFO

72     $path_info = $ENV{'PATH_INFO'};
73     $path_info =~ s/[<>!&~]//g;
74     $path_info =~ s/\.{2,}//g;      # don't allow uptree accesses
75     $path_info =~ s/\/{2,}/\//g;
76     $path_info =~ /()/; # Clear $1
77     $path_info =~ s#/([^\/]*)/?##;

78     $valid = 0;
79     $suid = "";
80     if ( $ENV{'HTTP_COOKIE'} =~ /SUID2\s*=\s*(\w*)/ ) {
81         $suid = $1;
82         $suid =~ s/\\s*/g;

83         ($user_id, $timestamp, $usagestamp, $usage, $valid) =
&decode_suid($suid);

```

```

84      # Some of the cookies are cached in a database.
85      # This code is intended to randomly audit the cookie data
86      # to make sure it isn't corrupt or counterfeited.

87      sysopen(AUDIT_DB, "auth.dat", 2) or die("Can't open authorization
database!");
88      my $hash = &hash($user_id);
89      sysseek( AUDIT_DB, $hash*$RECORD_LENGTH, 0 );
90      sysread( AUDIT_DB, my $record, $RECORD_LENGTH-1 ); # skip newline
91      (my $prev_user_id, my $prev_timestamp, my $prev_usagestamp, my
$prev_usage, my $prev_valid) =
92          &decode_suid($record);
93      close(AUDIT_DB);

94      if ( $prev_valid && ($prev_user_id eq $user_id) &&
95          ($prev_usagestamp eq $usagestamp) && ($usage < $prev_usage) ) {
96          $valid = 0; # Apparently the cookie is fake
97      }

98      # If more than 24 hours (or some other time limit) has passed,
99      # reset the usage count
100     if ( (time()-$usagestamp) > $USAGE_INTERVAL ) {
101         $usagestamp = time();
102         $usage = -1;
103     }

104     if ($usage > $USAGE_MAX) {
105         $valid = 0;
106     }

107     # DOS inhibition
108     if ( !( $ENV{"HTTP_USER_AGENT"} =~ /Mozilla/ ) ) {
109         sleep(15+rand(10)); return 0; }
110     else { # No cookie - create a new one
111         (my $ip = $ENV{"REMOTE_ADDR"}) =~ s/\./:/g;
112         (my $a, my $b, my $c, my $d) = split(":", $ip);
113         $ip = (((($a*256+$b)*256+$c)*256+$d);
114         my $rand = int(rand(2**24));
115         $user_id = sprintf("Z%08x%06x", $ip, $rand);

116         $usagestamp = time();
117         $usage = -1;
118     }

119     $new_suid = &encode_suid($user_id, time(), $usagestamp, $usage+1);

120     # Expire cookie in 90 days
121     $cookie = "Set-Cookie: SUID2=$new_suid;
expires=".&RFC950date(time()+7*24*3600)."; path=/cgi-bin/$PROJECT/;
domain=$HOST_IP\n";

122     if ($debug) { print "Using PATH_INFO...\n"; }

123     if ($debug) { print "SUID: ", $suid, " ", "PATH_INFO: ", $path_info, "\n"; }
124     1;

host.pl

```

```

1      #!/usr/bin/perl
2      #Copyright (C) 1997-2000 Rode Consulting, Inc.
3      #All rights reserved

4      use strict;

5      use vars qw($DOMAIN_NAME $HOST_NAME $HOST_IP $HOST_PREFIX
$HOST_CGI_PREFIX);
6      use vars qw($HOST_PREFIX_JAVA $HOST_CGI_PREFIX_JAVA);
7      use vars qw($BODY $NOCACHE_HEADER);
8      use vars qw(%HTML_MACROS);

9      $DOMAIN_NAME="circuitsim.com";
10     $HOST_NAME = "www.". $DOMAIN_NAME;
11     $HOST_IP = "24.218.251.135";

12     my $PREFIX = "Stackup/patent";
13     $HOST_PREFIX = "http://$HOST_NAME/$PREFIX/";
14     $HOST_CGI_PREFIX = "http://$HOST_NAME/cgi-bin/$PREFIX/";
15     $HOST_PREFIX_JAVA = "http://$HOST_IP/$PREFIX/";
16     $HOST_CGI_PREFIX_JAVA = "http://$HOST_IP/cgi-bin/$PREFIX/";
17     $BODY = '<BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF"
VLINK="#AA00AA" ALINK="#FF0000">';
18     $NOCACHE_HEADER = "Pragma: no-cache\nExpires: Mon, 01 Jul 1996 00:00:00
GMT\n";

19     $HTML_MACROS{"DOMAIN_NAME"} = '$DOMAIN_NAME';
20     $HTML_MACROS{"HOST_NAME"} = '$HOST_NAME';
21     $HTML_MACROS{"HOST_IP"} = '$HOST_IP';
22     $HTML_MACROS{"HOST_PREFIX"} = '$HOST_PREFIX';
23     $HTML_MACROS{"HOST_CGI_PREFIX"} = '$HOST_CGI_PREFIX';
24     $HTML_MACROS{"HOST_PREFIX_JAVA"} = '$HOST_PREFIX_JAVA';
25     $HTML_MACROS{"HOST_CGI_PREFIX_JAVA"} = '$HOST_CGI_PREFIX_JAVA';
26     $HTML_MACROS{"BODY"} = '$BODY';
27     $HTML_MACROS{"TIME"} = '$TIME';
28     $HTML_MACROS{"SESSIONID"} = '$session_id';
29     $HTML_MACROS{"SESSIONIDMACRO"} = '$session_id_macro';
30     1;

```

makeauth.pl

```

1      #!/usr/bin/perl

2      require("project.pl");

3      my $nothing = "";
4      for (my $i=0; $i<$RECORD_LENGTH-1; $i++) {
5          $nothing .= " ";
6      }

7      open(AUTH, ">auth.dat");

8      for (my $i=0; $i<$NUM_RECORDS; $i++) {
9          print AUTH $nothing, "\n";
10     }

```

project.pl

```

1      #!/usr/bin/perl
2      #Copyright (C) 1997 Rode Consulting, Inc.
3      #All rights reserved.

```

```

4      use strict;

5      use vars qw($PROJECT $USAGE_MAX $USAGE_INTERVAL $NUM_RECORDS
$RECORD_LENGTH);

6      $PROJECT = "Stackup";
7      $USAGE_MAX = 10;
8      $USAGE_INTERVAL = 12*3600;
9      $NUM_RECORDS = 10000;
10     $RECORD_LENGTH = 71;

11     use vars qw($DOMAIN_NAME $HOST_NAME $HOST_IP $HOST_PREFIX
$HOST_CGI_PREFIX);
12     use vars qw($HOST_PREFIX_JAVA $HOST_CGI_PREFIX_JAVA);
13     use vars qw($BODY $NOCACHE_HEADER $GS $JP2GIF $ALCHEMY);
14     use vars qw(%HTML_MACROS);

15     # Not a general purpose hash -
16     # intended to use positions 5-11 inclusive of user_id
17     sub hash() {
18         my $seed = pop @_;
19         srand( hex substr($seed,5,7) );
20         return int(rand($NUM_RECORDS));
21     }

22     require("host.pl");

23     1;

save.pl
1      #!/usr/bin/perl
2      # Copyright 1999-2000 Rode Consulting, Inc.
3      # All rights reserved

4      use strict;

5      require("getsessionid.pl");
6      use vars qw( $debug );
7      use vars qw ( $suid $user_id $timestamp $usage $valid $path_info
$cookie);

8      my $cgi_query;
9      read(STDIN, $cgi_query, $ENV{'CONTENT_LENGTH'});

10     print "Content-Type: text/plain\n";
11     print "Pragma: no-cache\nExpires: Mon, 01 Jul 1996 00:00:00 GMT\n";
12     print "\n";

13     if (length($user_id) > 0) {
14         'rm -f tmp/*$user_id*';
15         open(STUFF,">tmp/stuff".$user_id.sprintf("%04d",$usage+1));
16         print STUFF $cgi_query;
17         close(STUFF);
18         print "OK\n";
19     }
20     else { print "ERROR\n"; }

webdate.pl

```

```

1      #!/usr/bin/perl
2      #Copyright (C) 1997 Rode Consulting, Inc.
3      #All rights reserved.

4      use strict;

5      # GLOBALS
6      use vars qw( $debug );

7      sub zeropad2 {
8          my $number = pop(@_);
9          my $padded;
10         if ($number < 10) { $padded = "0".$number; }
11         else { $padded = "" . $number; }

12         $padded;
13     }

14     sub RFC950date {
15         my $time = pop(@_);
16         (my $sec,my $min,my $hour,my $mday,my $mon,my $year,my $wday,my
17         $yday,my $isdat) = gmtime($time);
18         if ($year < 1970) { $year += 1900; }
19         if ($year < 1970) { $year += 100; }

20         # $formatted='/usr/local/bin/date -d '$time seconds' '+%A, %d-%b-%y
21         &T GMT' --universal';
22         ("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday")[$wday
23         ].", " .
24         &zeropad2($mday) . "-" .
25         ("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec")[$mon
26         ] . "-" .
27         $year . " " .
28         &zeropad2($hour) . ":" . &zeropad2($min) . ":" . &zeropad2($sec) .
29         " GMT";
30     }

31     sub modified {
32         my $file = pop(@_);
33         (my $dev,my $ino,my $mode,my $nlink,my $uid,my $gid,my $rdev,my
34         $size,my $atime,my $mtime,my $ctime,my $blksize,my $blocks) = stat($file);

35         if ($debug) { print "Modified: ", $mtime, " ", time(), "\n"; print $^T;
36     }

37     &RFC950date($mtime);
38 }

39 1;

native_CrossSection.html
1      <HTML>
2      <HEAD>
3      <TITLE>EXPERIMENTAL PCB Cross Section Editor</TITLE>
4      </HEAD>
5      <BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF" VLINK="#AA00AA"
ALINK="#FF0000">

```

```

6      <APPLET CODE="CrossSection.class" CODEBASE="http://<HOST_IP>/Stackup/"
ARCHIVE="CrossSection.jar" WIDTH=1024 HEIGHT=540>
7      <PARAM NAME=addgeometry VALUE=ENABLE>
8      <PARAM NAME=userid VALUE="00000000">
9      </APPLET><BR><BR>

10     <!-- Instructions deleted -->

11     </BODY>
12     </HTML>
nav3_CrossSection.html
1     <HTML>
2     <HEAD>
3     <TITLE>EXPERIMENTAL PCB Cross Section Editor</TITLE>
4     </HEAD>
5     <BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF" VLINK="#AA00AA"
ALINK="#FF0000">

6     <APPLET CODE="CrossSection.class" CODEBASE="http://<HOST_IP>/Stackup/"
ARCHIVE="CrossSection.zip" WIDTH=1024 HEIGHT=540>
7     <PARAM NAME=addgeometry VALUE=ENABLE>
8     <PARAM NAME=userid VALUE="00000000">
9     </APPLET><BR><BR>

10    <!-- Instructions deleted -->

11    </BODY>
12    </HTML>
../../html/Stackup/declined.html
1     <HTML>
2     <HEAD><TITLE>Declined</TITLE></HEAD>
3     <BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF" VLINK="#AA00AA"
ALINK="#FF0000">

4     Sorry you feel that way...

5     </BODY>
6     </HTML>
../../html/Stackup/error.html
1     <HTML>
2     <HEAD>
3     <TITLE>Extracting geometry...</TITLE>
4     <SCRIPT LANGUAGE="JavaScript"><!--
5         badbrowser = ((navigator.appName == "Microsoft Internet Explorer")
&& (parseInt(navigator.appVersion) == 4) && (navigator.appVersion.indexOf("Mac") >
0));
6         if (badbrowser) this.focus();
7         if (!badbrowser && window.focus) this.focus();
8         //--></SCRIPT>
9     </HEAD>

10    <BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF" VLINK="#AA00AA"
ALINK="#FF0000">
11    <TABLE VALIGN=CENTER WIDTH="100%" HEIGHT="100%"><TR><TD VALIGN=CENTER
ALIGN=CENTER><CENTER>
12    <H2>We're sorry an ERROR occurred in processing your data.<BR>
13    Please try again later.</H2>

```



```

14      <H3><I>Cookies must be enabled!</I></H3>
15      </CENTER></TD></TR></TABLE>
16      </BODY>

17      </HTML>

.../..html/Stackup/overuse.html
1      <HTML>
2      <HEAD>
3      <TITLE>Extracting geometry...</TITLE>
4      <SCRIPT LANGUAGE="JavaScript"><!--
5          badbrowser = ((navigator.appName == "Microsoft Internet Explorer")
&& (parseInt(navigator.appVersion) == 4) && (navigator.appVersion.indexOf("Mac") >
0));
6          if (badbrowser) this.focus();
7          if (!badbrowser && window.focus) this.focus();
8      //--></SCRIPT>
9      </HEAD>

10      <BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF" VLINK="#AA00AA"
ALINK="#FF0000">
11      <TABLE VALIGN=CENTER WIDTH="100%" HEIGHT="100%"><TR><TD VALIGN=CENTER
ALIGN=CENTER><CENTER>
12      <H2>We're sorry but you've exceeded your<BR>extraction quota for
today.<BR><BR>
13      Please try again tomorrow.</H2>
14      </CENTER></TD></TR></TABLE>
15      </BODY>

16      </HTML>

```